In the United States Patent and Trademark Office

Board of Patent Appeals and Interferences

Appeal Brief

In re the Application of:

S. S. Lightstone, et al.
Serial No. 09/627,662
Filed: July 28, 2000
Attorney Docket No. CA990022US1

HEURISTIC-BASED CONDITIONAL DATA INDEXING

Submitted by:

Rabindranath Dutta
Reg. No.: 51,010
Konrad, Raynes & Victor LLP
315 So. Beverly Dr., Ste. 210
Beverly Hills CA 90212
(310) 557-2292
(310) 556-7984 (fax)

TABLE OF CONTENTS

page

I. Real Party in Interest 1

II. Related Appeals, Interferences, and Judicial Proceedings 1

III. Status of the Claims 1

IV. Status of Amendments 1

V. Summary of the Invention 2

VI. Grounds of Rejection 4

A. Ground of Rejection 1: The indefiniteness rejection under 35 U.S.C. 112, second paragraph.

B. Ground of Rejection 2: The Obviousness Rejection Based on the Ponnekanti patent in view of the Watkins patent.

C. Ground of Rejection 3: The Obviousness Rejection Based on the Ponnekanti patent, in view of the Watkins patent, and further in view of the Huang patent.

D. Ground of Rejection 4: The Obviousness Rejection Based on the Ponnekanti patent, in view of the Watkins patent, and further in view of the Sundara patent.

VII. Argument 5

A. Ground of Rejection 1: The indefiniteness rejection under 35 U.S.C. 112, second paragraph.

B. Ground of Rejection 2: The Obviousness Rejection Based on the Ponnekanti patent in view of the Watkins patent.

1) Claims 27, 37, and 47
2) Claims 28, 38, and 48
3) Claims 29, 39, and 49
4) Claims 32, 42, and 52
5) Claims 33, 43, and 53
6) Claims 36, 46, and 56
7) Claims 58, 61, and 64
8) Other Remarks on Rejected Claims

C. Ground of Rejection 3: The Obviousness Rejection Based on the Ponnekanti patent, in view of the Watkins patent, and further in view of the Huang patent.
(1) Claims 57, 60, and 63

D. Ground of Rejection 4: The Obviousness Rejection Based on the Ponnekanti patent, in view of the Watkins patent, and further in view of the Sundara patent.
(1) Claims 59, 62, and 65

VIII. Conclusion 27

IX. Appendix A (Claims on Appeal) 28

X. Attachment - 1

X1. Attachment - 2

I. Real Party in Interest

The entire right, title and interest in this patent application is assigned to real party in interest International Business Machines Corporation.

II. Related Appeals, Interferences, and Judicial Proceedings

Appellant, Appellant's legal representative, and Assignee are not aware of any other prior or pending appeals, interferences, and judicial proceedings which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of the Claims

Claims 1-26 have been withdrawn with traversal in response to a restriction requirement.

Claims 27 - 65 are pending and have been rejected.

The final rejection of the claims is being appealed for pending claims 27-65.

IV. Status of Amendments

On May 25, 2004, Applicants submitted an amendment in response to the Final Office action dated March 25, 2004, in which Applicants amended the claims and added a new Figure to overcome the Examiner's objection to the figures. In an Advisory Action, dated July 9, 2004, the Examiner declined to enter the Amendment submitted on May 25, 2004.

On August 23, 2004, Applicants submitted an amendment subsequent to the submission of the Notice of Appeal dated July 26, 2004. The amendment added a new Figure to overcome the Examiner's objections and place the Application in a better condition for appeal. The

amendment also included arguments for withdrawing the objections based on 35 U.S.C. 112 to place the Application in better condition for appeal. The Examiner has not yet acted upon the amendment submitted on August 23, 2004.

Thus, the claims pending prior to the Final Office Action are the claims whose rejection is appealed because the Examiner has not entered the amendment submitted after the Final Office Action and the amendment submitted after the Notice of Appeal.

V. Summary of the Invention

The claims in one aspect is directed to a method, system, and program for updating an index on a database table when data is added to the table. Data records are received to load into the table. One of a first operation and a second operation is selected, wherein the first operation incrementally updates the index on the table as each received data record is added to the table and the second operation rebuilds the index from the table after all the received data records have been added to the table. The selected first operation or the selected second operation is used to incrementally update or rebuild the index respectively with the data from the received data records. Support for these claim requirements may be found in at least page 11, lines 19-24 and FIGs. 1, 2, 3 of the original specification.

In another aspect of the claims, a determination is made as to which of the first operation or second operation is more efficient, wherein the first or second operation determined to be more efficient is the selected operation used for updating the index with the received data records. Support for these claim requirements may be found on page 11, lines 25-28 of the original specification.

In yet another aspect of the claims, determining which operation is more efficient is a function of a percentage of the received data records to add to the table and characteristics of the index. In another aspect of the claims, the characteristics of the index used in determining which operation is more efficient comprise a size and complexity of the index. In a further aspect of the claims, the index comprises a binary tree structure, wherein a height of the index tree is indicative of the size and complexity of the index. Support for the claims may be found in at least page 12, lines 15-20 of the original specification.

In yet another aspect of the claims, determining which operation is more efficient further comprises considering at least one of a following factors: an estimated time required to extract index keys from the table; an estimated time to sort the index keys; and an estimated time to rebuild the index from the sorted keys. Support for the claims may be found in at least page 13, lines 3-17 of the original specification.

In a further aspect of the claims, a list of threshold values for different index sizes are maintained. The number of received data records are used to add to the table to determine a comparison value, wherein determining whether the first or second operation is more efficient is based on the comparison value and the threshold for the size of the index to be updated. In another aspect of the claims, the comparison value comprises the number of the received data records as a percentage of all data records in the table. In yet another aspect of the claims, the index comprises a binary tree and wherein the list of threshold values provides one threshold for each of a plurality of different height index binary trees, wherein the threshold selected for comparison with the comparison value is based on the height of the index to update. In another aspect of the claims, the first operation is more efficient if the comparison value is less than the

threshold value and wherein the second operation is more efficient if the comparison value is greater than the threshold value. The claims are supported by at least page 13, lines 18-22 of the original specification, and by at least page 12, line 13 - page 16, line 15 of the original specification.

In yet another aspect of the claims, selecting is performed by a heuristic determination function. In a further aspect of the claims, the heuristic determination function allows a user to specify a selection between an incremental update of the index and a full rebuild of the index. In a still further aspect of the claims, the heuristic determination function takes as input index meta-data. The claims are supported by at least page 10: lines 5-9; page 11: lines 19-24; and FIGs. 1, 2, 3 of the original specification. Additional support for the claims may also be found at least in original claims 1, 2, 5, 10, original claims 1-26, and in pages 2-16 of the original specification.

VI. Grounds of Rejection

A concise statement listing each ground of rejection presented for review is as follows:

A. Ground of Rejection 1: The indefiniteness rejection under 35 U.S.C. 112, second paragraph.

Claims 27, 37, 47 stand rejected under 35 U.S.C. 112, second paragraph for failing to particularly point out and distinctly claims the subject matter which applicant regards as the invention.

B. Ground of Rejection 2: The Obviousness Rejection Based on the Ponnekanti patent in view of the Watkins patent.

Claims 27-36, 37-46, 47-56, 58, 61, 64 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti (US 6,591,269) in view of Watkins (US 6,457,017).

C. Ground of Rejection 3: The Obviousness Rejection Based on the Ponnekanti patent, in view of the Watkins patent, and further in view of the Huang patent.

Claims 57, 60, 63 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti, in view of Watkins, and further in view of Huang (6,026,406).

D. Ground of Rejection 4: The Obviousness Rejection Based on the Ponnekanti patent, in view of the Watkins patent, and further in view of the Sundara patent.

Claims 59, 62, 65 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti, in view of Watkins, and further in view of Sundara (US 6,360, 228).

VII. Argument

A. Ground of Rejection 1: The indefiniteness rejection under 35 U.S.C. 112, second paragraph.

In the Final office action (dated 03/25/2004, paper number 15) the Examiner rejected claims 27, 37, and 47 under 35 U.S.C. 112 second paragraph by indicating that the word "respectively" gave an ambiguous meaning for each selected operation.

Claims 27, 37, 47 require using the selected first operation or second operation to

incrementally update or rebuild the index respectively with the data from the received data records. According to the Merriam-Webster online Dictionary (www.webster.com) "respectively" is an adverb that means "in the order given" (see attachment-1). Additionally, according to the American Heritage Dictionary of the English Language: Fourth Edition, 2000, respectively means "singly in the order designated or mentioned" (see attachment-2). According to these ordinary dictionary definitions of "respectively", the claims clearly require the selected first operation to incrementally update the index and the second operation to rebuild the index because the claims require that these operations be performed "respectively", i.e., where these operations are performed singly in the order mentioned.

Accordingly, it is respectfully submitted that the rejection of claims 27, 37, 47 under 35 U.S.C. 112, second paragraph, should be reversed because the requirements of the claims are definite and clear.

B. Ground of Rejection 2: The Obviousness Rejection Based on the Ponnekanti patent in view of the Watkins patent.

1) Claims 27, 37, and 47

The Examiner rejected as obvious [35 U.S.C. 103(a)] pending claims 27, 37, 47 as being unpatentable over Ponnekanti in view of Watkins. Applicants traverse.

These claims require using the selected first operation or second operation to incrementally update or rebuild the index respectively with the data from the received data records. Therefore, the claims require performing both of the following:

(a) using the selected first operation to incrementally update the index; and

(b) using the selected second operation to rebuild the index.

The cited Ponnekanti (col. 2: lines 58-60; col. 3: lines 4-30; col. 4: lines 20-26; col. 6: lines 1-45; col. 7: lines 50-63; col. 13: lines 35-67; col. 16: lines 2-67; col. 18: lines 37-67; item 160 in FIG. 1B), at pages 2, 5, 6 of the Final Office Action, discusses rebuilding an index, where the index is stored as a B+- Tree data structure. The cited Watkins (FIG. 2, col. 3, lines 8-26, and col. 5: lines 10-38), at pages 2, 5, 6 of the Final Office Action, discusses incrementally indexing managed files of a document management system. The cited Watkins also discusses a bulk file creator that allows incrementing indexing.

Although the cited references discuss indexing, neither the cited Ponnekanti nor the cited Watkins teach or suggest the claim requirement of selecting one of a first operation and second operation, and using the selected first operation to incrementally update the index and using the selected second operation to rebuild the index. The claims also require, rebuilding the index (i.e., a full rebuild of the index) after all the received data records have been added to the table. These specific claimed operations of selecting operations and then using a selected first operation to incrementally update the index, and using a selected second operation to rebuild the index are nowhere taught or suggested in the cited art.

In fact, the cited Ponnekanti repeatedly teaches away from the claim requirements of incremental updates. For example, the cited Ponnekanti in col. 11, lines 60-62 mentions that it is desirable to rebuild a few hundred pages in a transaction. Also, in col. 13, lines 1-15 the cited Ponnekanti discusses the advantages of rebuilding multiple pages in a single action. The cited Ponnekanti in col. 13, lines 11-15 discusses that besides saving log space, it has also been

observed that rebuilding multiple pages in a top action reduces the number of visits to level 1 pages significantly, reducing the calls to lock manager, latch manager, and the like. Additionally, in col. 19, lines 43-50, the cited Ponnekanti discusses that by rebuilding multiple leaf pages in each top action, the updates to level 1 pages can be batched resulting in significant reduction in logging an CPU time. Although the cited Ponnekanti discusses rebuilding pages, the cited Ponnekanti teaches away from being modified to perform incremental updates as the Examiner proposes. These cited col. 18, lines 52-54 of Ponnekanti mentions that incremental reorganization is difficult and mentions performing inline reorganization to avoid the problems of incremental reorganization. Thus, Ponnekanti teaches away from being modified to additionally perform incremental updates.

The cited Watkins discusses incrementally indexing managed files of a document management system. However, nowhere does the cited Watkins teach or suggest the claim requirement of using the selected first operation (incremental update) to incrementally update the index and using the second operation (rebuilding the index) to rebuild the index with the data from the data records. In fact, the cited Watkins teaches away from the claim requirements because the cited Watkins performs incremental updates and nowhere suggests rebuilding the index. The cited Watkins (col. 15: lines 53-55) mentions that the system dynamically updates metadata definitions on the fly without rebuilding the database or restarting the system. Further, the cited Watkins (col. 9: lines 52-56) discusses adding another attribute to the definition of, for example, a customer folder, without restarting or reorganizing the database. Therefore, although the cited Watkins mentions incremental updates, the cited Watkins teaches away from being modified to additionally perform rebuilding the index as the Examiner proposes.

In his "response to arguments" section of the Office Action dated 05/25/2004, the Examiner indicated that the cited Watkins discusses that a bulk file creator is used to delete, add, or update the indexes, which is an efficient method of file indexing. However, the cited Watkins discusses (col. 5: line 29-32) that "the bulk file creator 40 is a class which comprises a part of the infrastructure that allows incremental indexing by indexing only files that have been added or removed." The file indexing of Watkins cited by the Examiner is different from rebuilding the index as required by the claims because the file indexing of the cited Watkins is for incremental updates via incremental indexing and not for rebuilding the index. Therefore, the bulk file creator discussed in the cited Watkins is for incremental indexing and does not teach or suggest selecting the second operation to rebuild the index as required by the claims.

Neither the cited Ponnekanti nor the cited Watkins teach or suggest the claim requirements of selecting one of a first operation and a second operation, and using the selected first operation (incremental update) to incrementally update the index and using the second operation (rebuilding the index) to rebuild the index. While the cited Ponnekanti discusses rebuilding indexes, the cited Ponnekanti teaches away from being modified to additionally perform the claim requirement of incremental updates. The cited Watkins discusses incremental updates but teaches away from being modified to perform the claim requirement of rebuilding the index. The claimed combination of selecting the first operation to incrementally update the index and selecting the second operation to rebuild the index are neither taught nor suggested by either the cited Watkins or the cited Ponnekanti, either alone or in combination. Furthermore, the claims require selecting between incrementally updating the index and rebuilding the index. Nowhere do the cited references teach or suggest selecting between incrementally updating and

rebuilding as claimed. Instead, the cited Ponnekanti discusses rebuilding the index to the exclusion of incremental updates and the cited Watkins discusses incremental updates to the exclusion of rebuilding the index.

The Examiner concluded, without citing any supporting art or objective evidence. that it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of the cited Ponnekanti with the teaching of the cited Watkins so as to obtain an operation of incremental updates to the index in order to increase the database system response time (Final Office Action, dated 03/25/2004 Page 6). Applicants traverse because neither the cited Ponnekanti nor the cited Watkins teach or suggest the claim limitation of using the selected first operation to incrementally update the index and using the selected second operation to rebuild the index. Moreover, as discussed above, the cited references teach away from being modified to perform the additional operations the Examiner proposes.

Applicants further submit that the Examiner's proposed combination of Ponnekanti and Watkins is improper because the Examiner has not provided any objective teaching of a suggestion or proper motivation for the claimed requirement of using the selected first operation to incrementally update the index and using the selected second operation to rebuild the index. Instead, the Examiner makes a conclusory statement that the teachings of Ponnekanti and the teachings of Watkins could be modified as proposed to increase the database system response time. The Examiner has also made a conclusory statement that the combination would have increased the speed and system response time for retrieving records from the indexes maintained by the cited Watkins and Ponnekanti. Applicants submit that the Examiner provided motivations of increasing the database system response time and increasing the speed of retrieving any

particular data record of the table is too general and could lead to suggest many different solutions, and not the particular claimed one.

Moreover, the claim requirements of using the selected first operation to incrementally update the index and using the selected second operation to rebuild the index are not taught or suggested by the proposed combination of the Examiner. The Examiner does not explain how the general purpose of increasing the database response time or increasing the speed of retrieving any particular data record of a table would motivate and suggest to one skilled in the art to arrive at the claim requirements of using the selected first operation to incrementally update the index and using the selected second operation to rebuild the index. Further, the Examiner has not cited any objective source that suggests or provides motivation for arriving at the claim requirements of using the selected first operation to incrementally update the index and using the selected second operation to rebuild the index.

The recent U.S. Court of Appeals for the Federal Circuit ("Federal Circuit") decision in In re Lee, 61 USPQ2d 1430 (Fed. Cir. 2002) is particularly instructive as to why the Examiner's proposed combination of Watkins and Ponnekanti is improper. In Lee, the Federal Circuit emphasized that it is essential that the decision to combine references "must be based on objective evidence of record". Id. at 1433. The Federal Circuit said that authority is required and cannot be substituted with "'[c]ommon knowledge and common sense,' even if assumed to derive from the agency's expertise." Id. 1435. The Federal Circuit said that this means that the Examiner cannot rely on conclusory statements of motivation. Id. at 1434. ("Conclusory statements such as those here provided do not fulfill the agency's obligation").

Under Lee, the Examiner's motivation is conclusory because the Examiner has not

explained how the general purpose of increasing the database response time or increasing the speed of retrieving any particular data record of a table would motivate one skilled in the art to use the selected first operation to incrementally update the index and using the selected second operation to rebuild the index. Further, the Examiner's proffered motivation is not supported by objective evidence and appears to be based on "common knowledge", which under Lee is improper.

Applicants further submit that the Examiner is engaging in inappropriate use of hindsight in combining the cited Ponnekanti and the cited Watkins. The Federal Circuit has made clear that some objective teaching of the suggestion or motivation to combine prior art references is needed. In re Dembiczak, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999) ("Our case law makes clear that the best defense against the subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references.") Here, the Examiner is engaging in improper use of hindsight to justify the proposed modification because the Examiner has not provided any objective teaching to support the proposed combination and modification. The Examiner's motivation for the combination or modification comes from the claim language itself and not any objective evidence.

Still further, case law is very clear that just because one can readily modify the prior art to produce the claimed combination does not make the claimed combination obvious unless the Examiner can provide some objective reference teaching or suggesting the claimed combination. "The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." In re Gordon, 221

USPQ 1125, 1127 (Fed. Cir. 1984). Even modifications that may appear simple cannot be made unless the Examiner provides suggestion or motivation to make the apparent "simple" modification. According to the Federal Circuit,

> In a proper obviousness determination, "[w]hether the changes from the prior art are 'minor', . . . the changes must be evaluated in terms of the whole invention, including whether the prior art provides any teaching or suggestion to one of ordinary skill in the art to make the changes that would produce the patentee's . . . device." Northern Telecom, Inc. v. Datapoint Corp., 908 F.2d 931, 935, 15 USPQ2d 1321, 1324 (Fed. Cir.), cert. denied, 498 U.S. 920 (1990). This includes what could be characterized as simple changes, as in In re Gordon, 733 F.2d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984) (Although a prior art device could have been turned upside down, that did not make the modification obvious unless the prior art fairly suggested the desirability of turning the device upside down.).

In re Chu, 36 USPQ2d 1089, 1094 (Fed. Cir. 1995).

Here, the Examiner has not offered any objective evidence of a suggestion or motivation in the art to combine the teachings or modify the cited Watkins and the cited Ponnekanti to arrive at the claims requirements of using the selected first operation to incrementally update the index and using the selected second operation to rebuild the index. The fact that combination of Watkins and Ponnekanti is generally desirable is not sufficient to justify the Examiner's proposed combination. There is no suggestion in either the cited Ponnekanti or the cited Watkins to suggest the desirability of modifying the cited Ponnekanti or Watkins with the teachings of the other to arrive at the claim requirements. Just increasing the database response time as discussed by the Examiner is a general advantage that is present in many database systems and the cited Ponnekanti's discussion of the increasing the speed of retrieving any particular data record of a table does not refer or concern incremental updates as required by the claims. Rather, the cited Ponnekanti's discussion of increasing the speed of retrieving is with respect to rebuilding the

index. Hence, there is no teaching or suggestion in the cited Ponnekanti or the cited Watkins to arrive at the claimed combination.

Applicants further submit that the neither the cited Watkins nor the cited Ponnekanti suggests the claimed combination of selecting one of a first operation and second operation, wherein the first operation incrementally updates the index on the table as each received data record is added to the table and the second operation rebuilds the index from the table after all the received data records have been added to the table, and using the selected first operation to incrementally update the index with the data from the received data records, and using the selected second operation to rebuild the index with the data from the received data records.

Accordingly, neither the cited Ponnekanti nor the cited Watkins teach or suggest, either alone or in combination, performing both of the following as required by the claims: (a) using the selected first operation to incrementally update the index; and (b) using the selected second operation to rebuild the index.

For the above reasons, pending independent claims 27, 37, and 47 are patentable over the cited Ponnekanti and the cited Watkins because neither Ponnekanti nor Watkins teach or suggest all the claim limitations either alone or in combination.

Accordingly, it is respectfully submitted that the rejection of claims 27, 37, 47 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins should be reversed.

2) Claims 28, 38, and 48

Pending claims 28, 38, and 48 depends from claim 27, 37 and 47 respectively and further require determining which of the first operation or second operation is more efficient, wherein

the first or second operation determined to be more efficient is the selected operation used for updating the index with the received data records.

In rejecting claims 28, 38, and 48, the Examiner cited col. 13: lines 50-67 of the cited Ponnekanti as disclosing the claim requirement of determining which of the first operation or second operation is more efficient, wherein the first or second operation determined to be more efficient is the selected operation used for updating the index with the received data records. (Final Office Action, pg. 6) Applicants traverse.

The cited Ponnekanti (co. 13: lines 50-67) discusses updating the index by rebuilding the index via insert and delete operations. The first operation of the claim requirements is for incrementally updating the index and the second operation of the claim requirements is for rebuilding the index. Nowhere, does the cited Ponnekanti teach or suggest the first operation of incrementally updating the index.

For the above reasons, pending independent claims 28, 38, and 48 are patentable over the cited Ponnekanti and the cited Watkins, because neither the cited Ponnekanti nor the cited Watkins teach or suggest all the claim limitations either alone or in combination.

Accordingly, it is respectfully submitted that the rejection of claims 28, 38, and 48 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins should be reversed.

3) Claims 29, 39, and 49

Pending claims 29, 39, and 50 depend from claim 28, 38 and 48, respectively, and further require that determining which operation is more efficient is a function of a percentage of the received data records to add to the table and characteristics of the index.

The Examiner cite col. 8, lines 22-30, col. 9,lines 48-55, and col. 11, lines 15-28 as disclosing the claim requirement determining which operation is more efficient is a function of a percentage of the received data records to add to the table and characteristics of the index. (Final Office Action, pg. 6) Applicants traverse.

Nowhere does the cited Ponnekanti (col. 8, lines 22-30; col. 9: lines 48-55; col. 11: lines 15-28) teach or suggest the claim requirements of determining which operation is more efficient as a function of a percentage of the received data records to add to the table and characteristics of the index.

The cited col. 8, lines 22-30 of Ponnekanti discusses rebuilding an index in a B+- tree and nowhere teaches or suggests the claim requirement of determining between the relative efficiency of a first operation (incremental update) and a second operation (rebuilding the index after all data records have been added).

The cited col. 9, lines 48-55 of Ponnekanti discusses the likelihood of blocking which does not teach or suggest in any way the claim requirement of the determination of the relative efficiency of a first operation (incremental update) and a second operation (rebuilding the index after all data records have been added).

The cited col. 11, lines 15-28 of Ponnekanti discusses retraversal strategies of the B+- tree with no teaching or suggestion of the claim requirement of determining the relative efficiency of a first operation (incremental update) and a second operation (rebuilding the index after all data records have been added).

For the above reasons, pending dependent claims 29, 39, and 49 are patentable over the over the cited over the cited Ponnekanti and the cited Watkins, because neither the cited

Ponnekanti nor the cited Watkins teach or suggest all the claim limitations either alone or in combination.

Accordingly, it is respectfully submitted that the rejection of claims 29, 39, and 49 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins should be reversed.

4) Claims 32, 42, and 52

Pending claims 32, 42, and 52 depend from claims 28, 38, and 48 respectively and further require that determining which operation is more efficient further comprises considering at least one of a following factors: an estimated time required to extract index keys from the table, an estimated time to sort the index keys, and an estimated time to rebuild the index from the sorted keys.

The cited col. 2: lines 1-16, col. 6: lines 30-45, col. 13: lines 35-67, and col. 16: lines 2-67 of Ponnekanti as disclosing the claim requirement that determining which operation is more efficient comprises considering at least one of a following factors: an estimated time required to extract index keys from the table, an estimated time to sort the index keys, and an estimated time to rebuild the index from the sorted keys' and maintaining a list of threshold values for different index sizes. (Final Office Action, pg. 7) Applicants traverse.

Nowhere does the cited Ponnekanti (col. 2, lines 1-16; col. 6: lines 30-45; col. 13: lines 35-67; col. 16: lines 2-67) teach or suggest the claim requirements of determining which operation is more efficient by considering at least one of the following factors: an estimated time required to extract index keys from the table, an estimated time to sort the index keys, and an estimated time to rebuild the index from the sorted keys. Instead, the cited Ponnekanti discusses

rebuilding the index without incremental updates and does not teach or suggest the claim requirement of determining which operation (i.e., the first operation of incremental update or the second operation of rebuilding the index) is more efficient as required by the claims.

For the above reasons, pending dependent claims 32, 42, and 53 are patentable over the over the cited over the cited Ponnekanti and the cited Watkins, because neither the cited Ponnekanti nor the cited Watkins teach or suggest all the claim limitations either alone or in combination.

Accordingly, it is respectfully submitted that the rejection of claims 32, 42, and 53 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins should be reversed.

5) Claims 33, 43, and 53

Pending claims 33, 43, and 53 depends from claims 28, 38, and 48 respectively and further require maintaining a list of threshold values for different index sizes and using the number of received data records to add to the table to determine a comparison value, wherein determining whether the first or second operation is more efficient is based on the comparison value and the threshold for the size of the index to be updated.

The Examiner cited col. 2, lines 1-16, col. 6: lines 30-45, col. 13: lines 35-67, and col. 16: lines 2-67 of Ponnekanti as disclosing the claim requirement of determining which operation is more efficient by considering at least one of the following factors: an estimated time required to extract index keys from the table, an estimated time to sort the index keys, and an estimated time to rebuild the index from the sorted keys, and maintaining a list of threshold values for different index sizes. (Final Office Action, pg. 7) Applicants traverse.

Nowhere does the cited Ponnekanti (col. 2, lines 1-16; col. 6: lines 30-45; coil. 13: lines 35-67; col. 16: lines 2-67) teach or suggest the claim requirements of maintaining a list of threshold values for different index sizes and using the number of received data records to add to the table to determine a comparison value, wherein determining whether the first or second operation is more efficient is based on the comparison value and the threshold for the size of the index to be updated. Instead, the cited Ponnekanti discusses rebuilding the index without incremental updates and does not teach or suggest the claim requirement of determining which operation (i.e., the first operation of incremental update or the second operation of rebuilding the index) is more efficient based on the comparison value and the threshold for the size of the index to be updated as required by the claims.

For the above reasons, pending dependent claims 33, 43, and 55 are patentable over the over the cited over the cited Ponnekanti and the cited Watkins, because neither the cited Ponnekanti nor the cited Watkins teach or suggest all the claim limitations either alone or in combination.

Accordingly, it is respectfully submitted that the rejection of claims 33, 43, 55 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins should be reversed.

6) Claims 36, 46, and 56

Pending claims 36, 46, and 56 depend from claims 33, 43, and 53 respectively and further require that the first operation is more efficient if the comparison value is less than the threshold value and wherein the second operation is more efficient if the comparison value is greater than the threshold value.

The Examiner cited col. 7: lines 50-55; col. 5: lines 56-67; col. 7: lines 55-64; fig 2B; col. 7, lines 1-28; col. 8, lines 31-50; col. 13: lines 50-67 of Ponnekanti as disclosing the claim requirement that the number of the received data records is a percentage of all data records in the table, wherein the index comprises a binary tree and wherein the list of threshold values provides one threshold for each of a plurality of different height index binary trees, wherein the threshold selected for comparison with the comparison value is based on the height of the index to update and the threshold value and wherein the second operation is more efficient if the comparison value is greater than the threshold value. (Final Office Action, pg. 7)

Nowhere does the cited Ponnekanti teach or suggest the claim requirement that the first operation (i.e., incremental update) is more efficient if the comparison value is less than the threshold value and wherein the second operation (rebuilding the index) is more efficient if the comparison value is greater than the threshold value.

The cited Ponnekanti (col. 7: lines 50-55; col. 5: lines 56-67; col. 7: lines 55-64; fig 2B; col. 7, lines 1-28; col. 8, lines 31-50; col. 13: lines 50-67) discusses rebuilding the index without incremental updates. However, nowhere does the cited Ponnekanti teach or discuss the claim requirement that the first operation (i.e., incremental update) is more efficient if the comparison value is less than the threshold value and wherein the second operation (rebuilding the index) is more efficient if the comparison value is greater than the threshold value. In contrast, the cited Ponnekanti discusses efficiency issues in rebuilding the index without incremental updates.

For the above reasons, pending dependent claims 33, 43, and 55 are patentable over the over the cited over the cited Ponnekanti and the cited Watkins, because neither the cited Ponnekanti nor the cited Watkins teach or suggest all the claim limitations either alone or in

combination.

Accordingly, it is respectfully submitted that the rejection of claims 33, 43, and 55 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins should be reversed.

7) Claim 58, 61, and 64

Claim 58 depends on claim 57, wherein the heuristic determination function allows a user to specify a selection between an incremental update of the index and a full rebuild of the index.

The Examiner rejected claims 58, 61, and 64 as obvious [35 U.S.C. 103(a)] over Ponnekanti in view of Watkins. The Examiner did not cite any specific sections of the cited art in rejecting these claims.

Accordingly, it is respectfully submitted that the rejection of claims 58, 61, and 64 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins should be reversed.

8) Claims 28-36, 38-46, 48-56, 58, 61, 64

Additionally, claims 28-36, 38-46, 48-56, 58, 61, 64 depend directly or indirectly on the pending independent claims 27, 37, or 47. Applicants submit that these claims are patentable over the cited art because they depend from claims 27, 37, or 47 which are patentable over the cited art for the reason discussed above, and because the combination of the limitations in the dependent claims and the base and intervening claims from which claims 28-36, 38-46, 48-56, 58, 61, 64 depend provide further grounds of distinction over the cited art .

Accordingly, it is respectfully submitted that the rejection of claims 27-36, 37-46, 47-56,

58, 61, 64 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins should be reversed.

C. Ground of Rejection 3: The Obviousness Rejection Based on the Ponnekanti patent, in view of the Watkins patent, and further in view of the Huang patent.

(1) Claims 57, 60, and 63

Claims 57, 60, and 63 depend on claims 27, 37 and 47 respectively wherein selecting is performed by a heuristic determination function.

The Examiner has rejected [Page 11, Item 8 of the office action] as obvious (35 U.S.C. §103) pending claims 57, 60, 63 as being unpatentable over Ponnekanti, in view of Watkins, and in view of Huang (US 6,026,406).

The Examiner acknowledges that the cited Ponnekanti and the cited Watkins do not teach or suggest a heuristic determination function (office action: page 11). The cited Huang (col. 2: lines 34-42, col. 3: lines 12-20; col. 6: lines 8-21) discusses using a user defined cost threshold value (Huang: col. 6: lines 8-11) to perform index maintenance operation. Nowhere does the cited Huang teach or suggest using a heuristic determination function. Instead, the cited Huang is using a user defined cost threshold value. Nowhere does the cited Huang teach or suggest that the user defined cost threshold value is determined heuristically or is a heuristic determination function as required by the claims.

Moreover, Huang teaches away from the claim requirements of selection using a heuristic determination function because the selection discussed in the cited Huang is for an operation that is different from that required by the claim limitations of selecting one of a first operation and

second operation, wherein the first operation incrementally updates the index and the second operation rebuilds the index from the table after all the received data records have been added to the table. Instead, the cited Huang discusses the selection for batch processing of updates or for per update approaches. Thus, nowhere does the cited Huang anywhere teach or suggest using a heuristic determination function to choose between full rebuild as discussed in the cited Ponnekanti and the incremental update as discussed in the cited Watkins.

Applicants submit that the Examiner's proposed modification of Huang is improper because the Examiner has not provided any objective teaching of a suggestion or proper motivation to indicate that the cited Huang, or the cited Ponnekanti, or the cited Watkins, suggest the desirability of the claimed combination of selecting, by a heuristic function, one of a first operation and second operation, wherein the first operation incrementally updates the index on the table as each received data record is added to the table and the second operation rebuilds the index from the table after all the received data records have been added to the table.

The recent U.S. Court of Appeals for the Federal Circuit ("Federal Circuit") decision in In re Lee, 61 USPQ2d 1430 (Fed. Cir. 2002) is particularly instructive as to why the Examiner's proposed combination of Watkins, Ponnekanti, and Huang is improper. In Lee, the Federal Circuit emphasized that it is essential that the decision to combine references "must be based on objective evidence of record". Id. at 1433. The Federal Circuit said that authority is required and cannot be substituted with "'[c]ommon knowledge and common sense,' even if assumed to derive from the agency's expertise." Id. 1435. The Federal Circuit said that this means that the Examiner cannot rely on conclusory statements of motivation. Id. at 1434. ("Conclusory statements such as those here provided do not fulfill the agency's obligation").

Under Lee, the Examiner's motivation is conclusory because the Examiner has not explained how the general purpose of increasing the speed of retrieving a particular data record of a table and increasing database system response time as mentioned by the Examiner would motivate one skilled in the art to arrive at the claim requirements for selecting, by a heuristic function, one of a first operation and second operation, wherein the first operation incrementally updates the index on the table as each received data record is added to the table and the second operation rebuilds the index from the table after all the received data records have been added to the table. Further, the Examiner's proffered motivation is not supported by objective evidence and appears to be based on "common knowledge", which under Lee is not permissible.

Increasing the speed of retrieving a particular data record of a table and increasing database system response time as mentioned by the Examiner are inadequate motivation for the combination because such increase in speed and system response can be used for improving any database system. Additionally, the Examiner suggested motivation is using hindsight to combine the cited Huang, the cited Watkins, and the cited Ponnekanti to arrive at the claim requirement.

Additionally, pending dependent claims 57, 60, and 63 depend on the pending independent claims 27, 37, and 47 respectively. Applicants submit that pending claims 57, 60, and 63 are patentable over the cited art because they depend from claims 27, 37, and 47 respectively which are patentable over the cited art for the reason discussed above, and because the combination of the limitations in the dependent claims 57, 60, and 63 and the base and intervening claims from which they depend provide further grounds of distinction over the cited art.

For the above reasons, pending dependent claims 57, 60, and 63 are patentable over the over the cited over the cited Ponnekanti, the cited Watkins, and the cited Huang because neither the cited Ponnekanti nor the cited Watkins nor the cited Huang teach or suggest all the claim limitations either alone or in combination.

Accordingly, it is respectfully submitted that the rejection of claims 57, 60, 63 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins, and further in view of Huang should be reversed.

D. Ground of Rejection 4: The Obviousness Rejection Based on the Ponnekanti patent, in view of the Watkins patent, and further in view of the Sundara patent.

Claims 59, 62, 65 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti, in view of Watkins, and further in view of Sundara (US 6,360, 228)

(1) Claims 59, 62, and 65

Claim 59, 62, and 65 depend on claims 57, 60, and 63 respectively wherein the heuristic determination function takes as input index meta-data.

The Examiner rejected as obvious (35 U.S.C. §103) pending claims 59, 62, 65 as being unpatentable over Ponnekanti, in view Watkins, and in view of Sundara (US 6,360,228). The Examiner mentions that the cited Ponnekanti discloses rebuilding an index, the cited Watkins discloses incrementally updating an index, and the cited Sundara discloses user input or metadata defining an index type, where the metadata describes the index domain. [Office Action dated 3/25/2004: Page 12: Item 10] Applicants traverse.

Nowhere does the cited Sundara (col. 1, lines 54-67; col. 8: lines 2-42) teach or suggest

the claim requirement of the heuristic determination function. Additionally, nowhere does the cited Sundara teach or suggest the claim requirement that the heuristic determination function takes as input index meta-data.

Col. 1, lines 54-67 of the cited Sundara discusses user supplied meta-data that defines an index type. Col. 8 lines 2-42 discusses breaking DDL operations into transactions and preventing a transactional context change for DML operations. Although the cited Sundara discusses user supplied meta-data, nowhere does the cited Sundara, the cited Ponnekanti nor the cited Watkins teach or suggest the claim requirement that the heuristic determination function takes as input index meta-data.

The Examiner rejected claims 59, 62, 65 in view of Sundara, Ponnekanti and Watkins. (Final Office Action, pg. 12-14) Claims 59, 62, and 65 depend on claims 57, 60 and 63 respectively that were rejected by the Examiner in view of Huang, in view of Ponnekanti, and in view of Watkins.

Additionally, pending dependent claims 59, 62, and 65 depend directly or indirectly on the pending independent claims 27, 37, and 47 respectively. Applicants submit that pending claims 59, 62, and 65 are patentable over the cited art because they depend from claims 27, 37, and 47 respectively which are patentable over the cited art for the reason discussed above, and because the combination of the limitations in the dependent claims 59, 62, and 65 and the base and intervening claims from which they depend provide further grounds of distinction over the cited art.

For the above reasons, pending dependent claims 59, 62, 65 are patentable over the over the cited over the cited Ponnekanti, the cited Watkins, and the cited Sundara because neither the

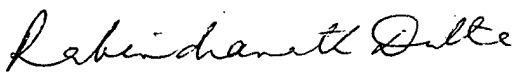
cited Ponnekanti nor the cited Watkins nor the cited Sundara teach or suggest all the claim limitations either alone or in combination.

Accordingly, it is respectfully submitted that the rejection of claims 59, 62, 65 under 35 U.S.C. 103(a) as being unpatentable over Ponnekanti in view of Watkins, and further in view of Sundara should be reversed.

VIII. Conclusion

Each of the rejections set forth in the final Office Action is improper and should be reversed.

Respectfully submitted,



Rabindranath Dutta
Reg. No. 51,010

Dated: October 26, 2004

Direct All Correspondence to:
Rabindranath Dutta
Konrad Raynes & Victor LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, California 90212
Tel: 310-557-2292
Fax: 310-556-7984

IX. Appendix A

The claims on appeal are as follows:

27. (Previously presented) A method for updating an index on a database table when data is added to the table, comprising:

receiving data records to load into the table;

selecting one of a first operation and second operation, wherein the first operation incrementally updates the index on the table as each received data record is added to the table and the second operation rebuilds the index from the table after all the received data records have been added to the table; and

using the selected first operation or second operation to incrementally update or rebuild the index respectively with the data from the received data records.

28. (Previously presented) The method of claim 27, further comprising:

determining which of the first operation or second operation is more efficient, wherein the first or second operation determined to be more efficient is the selected operation used for updating the index with the received data records.

29. (Original) The method of claim 28, wherein determining which operation is more efficient is a function of a percentage of the received data records to add to the table and characteristics of the index.

30. (Original) The method of claim 29, wherein the characteristics of the index used in

determining which operation is more efficient comprise a size and complexity of the index.

31. (Original) The method of claim 30, wherein the index comprises a binary tree structure, and wherein a height of the index tree is indicative of the size and complexity of the index.

32. (Original) The method of claim 28, wherein determining which operation is more efficient further comprises considering at least one of a following factors: an estimated time required to extract index keys from the table; an estimated time to sort the index keys; and an estimated time to rebuild the index from the sorted keys.

33. (Original) The method of claim 28, further comprising:

maintaining a list of threshold values for different index sizes; and

using the number of received data records to add to the table to determine a comparison value, wherein determining whether the first or second operation is more efficient is based on the comparison value and the threshold for the size of the index to be updated.

34. (Original) The method of claim 33, wherein the comparison value comprises the number of the received data records as a percentage of all data records in the table.

35. (Original) The method of claim 34, wherein the index comprises a binary tree and wherein the list of threshold values provides one threshold for each of a plurality of different

height index binary trees, wherein the threshold selected for comparison with the comparison value is based on the height of the index to update.

36. (Original) The method of claim 33, wherein the first operation is more efficient if the comparison value is less than the threshold value and wherein the second operation is more efficient if the comparison value is greater than the threshold value.

37. (Previously presented) A system for updating an index on a database table when data is added to the table, comprising:

a database system including the table and the index on the table;

means for receiving data records to load into the table;

means for selecting one of a first operation and second operation, wherein the first operation incrementally updates the index on the table as each received data record is added to the table and the second operation rebuilds the index from the table after all the received data records have been added to the table; and

means for using the selected first operation or second operation to incrementally update or rebuild the index respectively with the data from the received data records.

38. (Previously presented) The system of claim 37, further comprising:

means for determining which of the first operation or second operation is more efficient, wherein the first or second operation determined to be more efficient is selected to use for updating the index with the received data records.

39. (Original) The system of claim 38, wherein the means for determining which operation is more efficient is a function of a percentage of the received data records to add to the table and characteristics of the index.

40. (Original) The system of claim 39, wherein the characteristics of the index used in determining which operation is more efficient comprise a size and complexity of the index.

41. (Original) The system of claim 40, wherein the index comprises a binary tree structure, and wherein a height of the index tree is indicative of the size and complexity of the index.

42. (Original) The system of claim 38, wherein the means for determining which operation is more efficient further comprises considering at least one of a following factors: an estimated time required to extract index keys from the table; an estimated time to sort the index keys; and an estimated time to rebuild the index from the sorted keys.

43. (Original) The system of claim 38, further comprising:

means for maintaining a list of threshold values for different index sizes; and

means for using the number of received data records to add to the table to determine a comparison value, wherein determining whether the first or second operation is more efficient is based on the comparison value and the threshold for the size of the index to be updated.

44. (Original) The system of claim 43, wherein the comparison value comprises the number of the received data records as a percentage of all data records in the table.

45. (Original) The system of claim 44, wherein the index comprises a binary tree and wherein the list of threshold values provides one threshold for each of a plurality of different height index binary trees, wherein the threshold selected for comparison with the comparison value is based on the height of the index to update.

46. (Original) The system of claim 43, wherein the first operation is more efficient if the comparison value is less than the threshold value and wherein the second operation is more efficient if the comparison value is greater than the threshold value.

47. (Previously presented) A program for updating an index on a database table when data is added to the table, wherein the program is embedded in a computer readable medium and capable of causing a computer to perform:

receiving data records to load into the table;

selecting one of a first operation and second operation, wherein the first operation incrementally updates the index on the table as each received data record is added to the table and the second operation rebuilds the index from the table after all the received data records have been added to the table; and

using the selected first operation or second operation to incrementally update or rebuild the index respectively with the data from the received data records.

48. (Previously presented) The program of claim 47, wherein the program is further capable of causing the processor to perform:

determining which of the first operation or second operation is more efficient, wherein the first or second operation determined to be more efficient is selected to use for updating the index with the received data records.

49. (Original) The program of claim 48, wherein determining which operation is more efficient is a function of a percentage of the received data records to add to the table and characteristics of the index.

50. (Original) The program of claim 49, wherein the characteristics of the index used in determining which operation is more efficient comprise a size and complexity of the index.

51. (Original) The program of claim 50, wherein the index comprises a binary tree structure, and wherein a height of the index tree is indicative of the size and complexity of the index.

52. (Original) The program of claim 49, wherein determining which operation is more efficient further comprises considering at least one of a following factors: an estimated time required to extract index keys from the table; an estimated time to sort the index keys; and an estimated time to rebuild the index from the sorted keys.

53. (Original) The program of claim 49, wherein the program is further capable of causing the processor to perform:

maintaining a list of threshold values for different index sizes; and

using the number of received data records to add to the table to determine a comparison value, wherein determining whether the first or second operation is more efficient is based on the comparison value and the threshold for the size of the index to be updated.

54. (Original) The program of claim 53, wherein the comparison value comprises the number of the received data records as a percentage of all data records in the table.

55. (Original) The program of claim 54, wherein the index comprises a binary tree and wherein the list of threshold values provides one threshold for each of a plurality of different height index binary trees, wherein the threshold selected for comparison with the comparison value is based on the height of the index to update.

56. (Original) The program of claim 53, wherein the first operation is more efficient if the comparison value is less than the threshold value and wherein the second operation is more efficient if the comparison value is greater than the threshold value.

57. (Previously presented) The method of claim 27, wherein selecting is performed by a heuristic determination function.

58. (Previously presented) The method of claim 57, wherein the heuristic determination function allows a user to specify a selection between an incremental update of the index and a full rebuild of the index.

59. (Previously presented) The method of claim 57, wherein the heuristic determination function takes as input index meta-data.

60. (Previously presented) The system of claim 37, wherein the means for selecting comprises a heuristic determination function.

61. (Previously presented) The system of claim 60, wherein the heuristic determination function allows a user to specify a selection between an incremental update of the index and a full rebuild of the index.

62. (Previously presented) The system of claim 60, wherein the heuristic determination function takes as input index meta-data.

63. (Previously presented) The program of claim 47, wherein selecting is performed by a heuristic determination function.

64. (Previously presented) The program of claim 63, wherein the heuristic determination function allows a user to specify a selection between an incremental update of the

index and a full rebuild of the index.

65. (Previously presented) The program of claim 63, wherein the heuristic determination function takes as input index meta-data.

ATTACHMENT - 1

HOME
PREMIUM SERVICES
M-WCollegiate.com
M-WUnabridged.com
Britannica.com
Multi-User Licenses
DOWNLOADS
WORD OF THE DAY
WORD GAMES
WORD FOR THE WISE
ONLINE STORE
HELP
Merriam-Webster Inc. Company Information

Merriam-Webster Online Dictionary

Thesaurus

3 entries found for **respectively**.
To select an entry, click on it.

respectively
chlorophyll
vitamin K
Go

Main Entry: **re·spec·tive·ly**
Function: *adverb*
1 : in particular : SEPARATELY <could not recognize the solutions as salty or sour, *respectively*>
2 : in the order given <Mary and Anne were *respectively* 12 and 16 years old>

For **More Information on "respectively" go to Britannica.com**

Get the **Top 10 Search Results for "respectively"**

Pronunciation Symbols

ATTACHMENT - 2

The American Heritage® Dictionary of the English Language: Fourth Edition. 2000.

respectively

SYLLABICATION: re·spec·tive·ly

PRONUNCIATION: rĭ-spĕk'tĭv-lē

ADVERB: Singly in the order designated or mentioned: *I'm referring to each of you respectively.*

CONTENTS · INDEX · ILLUSTRATIONS · BIBLIOGRAPHIC RECORD
< respective respell >